Amendments to the Claims

1. (currently amended) A method for masking ~~version~~ differences among a plurality of ~~applications~~ servers providing similar services over a network, the method comprising the steps of:

receiving, at an application switching component from a requesting process, a request for a service among the similar services, wherein the request includes data indicating a particular service extension is mandatory;

wherein the application switching component is a process that switches among the plurality of servers;

sending the request to a first ~~application~~ server of the plurality of ~~applications~~ servers;

receiving, at the application switching component in response to sending the request to the first ~~application~~ server, error data that indicates the particular service extension is not available at the first ~~application~~ server, wherein the error data is not sent to the requesting process; and

in response to receiving the error data, sending the request from the application switching component to a second ~~application~~ server of the plurality of ~~applications~~ servers, wherein the second ~~application~~ server is different from the first ~~application~~ server.

2. (currently amended) A method as recited in Claim 1, further comprising the steps of:

determining at the application switching component whether none of the plurality of ~~applications~~ servers has the particular service extension; and

if it is determined that none has the particular service extension, then sending, to the requesting process, error data indicating the particular service extension is not available.

3. (currently amended) A method as recited in Claim 2, further comprising the step of:

if it is determined that at least one of the plurality of ~~applications~~ servers has the particular service extension, then not sending, to the requesting process, error data indicating the particular service extension is not available.

4. (original) A method as recited in Claim 1, wherein:

said step of receiving the request for services is performed by receiving a request formatted according to the Simple Object Access Protocol (SOAP); and

the data indicating the particular service extension is mandatory is included in a mustUnderstand attribute associated with the particular service extension.

5. (original) A method as recited in Claim 1, further comprising the step of:

sending to the requesting process advertising data indicating that another service that appears to be assembled out of the similar services is available at a network address of the application-switching component.

6. (original) A method as recited in Claim 1, further comprising the steps of:

determining at the application switching component whether a timeout period has occurred; and

if the timeout period has occurred, then sending, to the requesting process, error data indicating that the particular service extension is not available.

7. (currently amended) A computer-readable medium carrying one or more sequences of instructions for masking ~~version~~ differences among a plurality of ~~applications~~ servers providing similar services over a network, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of [[;]]:

receiving, at an application switching component from a requested process, a request for a service among the similar services, wherein the request includes data indicating a particular service extension is mandatory;

sending the request to a first ~~application~~ server of the plurality of ~~applications~~ servers;

receiving, at the switching component in response to sending the request to the first ~~application~~ server, error data that indicates the particular service extension is not available at the first ~~application~~ server, wherein the error data is not sent to the requesting process; and in response to receiving the error data, sending the request to a second ~~application~~ server of the plurality of applications, wherein the second ~~application~~ server is different from the first ~~application~~ server.

8. (currently amended) A computer-readable medium as recited in Claim 7, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the steps of:

determining whether none of the plurality of ~~applications~~ servers has the particular service extension; and

if it is determined that none has the particular service extension, then sending, to the requesting process, error data indicating the particular service extension is not available.

9. (currently amended) A computer-readable medium as recited in Claim 8, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the step of:

if it is determined that at least one of the plurality of ~~applications~~ servers has the particular service extension, then not sending, to the requesting process, error data indicating the particular service extension is not available.

10. (original) A computer-readable medium as recited in Claim 7, wherein:

said step of receiving the request for services is performed by receiving a request formatted according to a Simple Object Access Protocol (SOAP); and

the data indicating the particular service extension is mandatory is included in a mustUnderstand attribute associated with the particular service extension.

11. (currently amended) A computer-readable medium as recited in Claim 7, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the step of:

sending to the requesting process advertising data indicating that another service that appears to be assembled out of the similar services [[are]] is available at a network address of the application switching component.

12. (currently amended) A computer-readable medium as recited in Claim [[6]] 7, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the steps of:

determining at the application switching component whether a timeout period has occurred; and

if the timeout period has occurred, then sending, to the requesting process, error data indicating that the particular service extension is not available.

13. (currently amended) An apparatus for masking ~~version~~ differences among a plurality of ~~applications~~ servers providing similar services over a network, comprising:

means for receiving from a requesting process a request for a service among the similar services, wherein the request includes data indicating a particular service extension is mandatory;

means for sending the request to a first ~~application~~ server of the plurality of ~~applications~~ servers;

means for receiving error data that indicates the particular service extension is not available at the first ~~application~~ server, wherein the error data is not sent to the requesting process; and

means for sending the request to a second ~~application~~ server of the plurality of ~~applications~~ servers in response to receiving the error data, wherein the second ~~application~~ server is different from the first ~~application~~ server.

14. (currently amended) An apparatus for masking ~~version~~ differences among a plurality of ~~applications~~ servers providing similar services over a network, comprising:

a network interface that is coupled to the network for receiving requests from a requesting process;

a processor connected to the network interface;

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

receiving from [[a]] the requesting process a request for a service among the similar services, wherein the request includes data indicating a particular service extension is mandatory;

sending the request to a first ~~application~~ server of the plurality of ~~applications~~ servers;

receiving error data that indicates the particular service extension is not available at the first ~~application~~ server, wherein the error data is not sent to the requesting process; and

sending the request to a second ~~application~~ server of the plurality of ~~applications~~ servers, wherein the second ~~application~~ server is different from the first ~~application~~ server.

15. (currently amended) A system for masking ~~version~~ differences among a plurality of ~~applications~~ servers providing similar services over a network, comprising:

a network interface that is coupled to the network for receiving requests from and sending responses to a requesting process;

a processor;

a plurality of stored sets of one or more sequences of instructions corresponding to the plurality of ~~applications~~ server providing similar services when executed by the processor; and

an application switching component connected between the processor and the network interface, wherein the application switching component switches among the plurality of servers and is configured to carry out the steps of:

receiving from the requesting process a request for a service among the similar services, wherein the request includes data indicating a particular service extension is mandatory;

sending the request to the processor while executing a first ~~application~~ server of the plurality of ~~applications~~ servers;

receiving error data that indicates the particular service extension is not available at the first ~~application~~ server, wherein the error data is not sent to the requesting process; and sending the request to the processor while executing a second ~~application~~ server of the plurality of ~~applications~~ servers, wherein the second ~~application~~ server is different from the first ~~application~~ server.

16. (currently amended) A method for masking ~~version~~ differences among a plurality of computer program servers providing similar services over a network, the method comprising the steps of:

receiving, at an application switching component from a requesting client, a SOAP message that includes a request for a service among the similar services, wherein the request includes a mustUnderstand attribute indicating a particular service extension is mandatory;

wherein the application switching component is a process that switches among the plurality of servers;

sending the request to a first server of the plurality of servers;

receiving, at the application switching component in response to sending the request to the first application, a mustUnderstand error value that indicates the particular service extension is not available at the first server, wherein the error data is not sent to the requesting client; ~~and~~

in response to receiving the error data, selecting a second server of the plurality of applications that is different from the first server; and

sending the request from the application switching component to the second server.

17. (original) A method as recited in Claim 16, further comprising the step of sending a mustUnderstand error message to the requesting client only when no second server is capable of selection.

18. (new) An apparatus as recited in Claim 13, further comprising:

means for determining at the application switching component whether none of the plurality of servers_has the particular service extension; and means for sending to the requesting process, only if it is determined that none has the particular service extension, error data indicating the particular service extension is not available.

19. (new) An apparatus as recited in Claim 13, wherein:

said means for receiving a request for a service includes means for receiving a request formatted according to the Simple Object Access Protocol (SOAP); and

a mustUnderstand attribute associated with the particular service extension includes the data indicating the particular service extension is mandatory.

20. (new) An apparatus as recited in Claim 13, further comprising:

means for sending to the requesting process advertising data indicating that another service that appears to be assembled out of the similar services is available at a network address of the apparatus.

21. (new) An apparatus as recited in Claim 13, further comprising:

means for determining whether a timeout period has occurred; and

means for sending, to the requesting process, if the timeout period has occurred, error data indicating that the particular service extension is not available.

22. (new) An apparatus as recited in Claim 14, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the steps of:

determining whether none of the plurality of servers has the particular service extension; and

if it is determined that none has the particular service extension, then sending, to the requesting process, error data indicating the particular service extension is not available.

23. (new) An apparatus as recited in Claim 14, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the step of:

if it is determined that at least one of the plurality of servers has the particular service extension, then not sending, to the requesting process, error data indicating the particular service extension is not available.

24. (new) An apparatus as recited in Claim 14, wherein:

said step of receiving the request for services is performed by receiving a request formatted according to a Simple Object Access Protocol (SOAP); and

a mustUnderstand attribute associated with the particular service extension includes the data indicating the particular service extension is mandatory.

25. (new) An apparatus as recited in Claim 14, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the step of:

sending to the requesting process advertising data indicating that another service that appears to be assembled out of the similar services is available at a network address of the apparatus.

26. (new) An apparatus as recited in Claim 14, wherein execution of the one or more sequences of instructions by one or more processors further causes the one or more processors to perform the steps of:

determining whether a timeout period has occurred; and

if the timeout period has occurred, then sending, to the requesting process, error data indicating that the particular service extension is not available.